

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Mobile TV

Cristhian David Santo Caldeira

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Prof.a Maria Teresa de Andrade

20 de Julho de 2017

Resumo

Com o evoluir da tecnologia evoluem as formas de estar na vida, como tal surge a necessidade de adaptar hábitos convencionais às lides modernas.

Desta forma deparamos-nos com este projeto desenvolvido no âmbito do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto, que visa concretizar uma alternativa moderna à convencional e por vezes ultrapassada televisão.

Para alcançar tal solução foi desenvolvido um estudo sobre Televisão e as tecnologias possíveis e usadas para a difusão do seu sinal, com o qual se conclui que o HTTP *Streaming* por meio do MPEG-DASH seria a hipótese que melhor se adequa.

Assim foi desenvolvido um conjunto de trabalhos práticos, focados na concessão de uma aplicação Android, que possibilite a interação com o utilizador e ver televisão através de dispositivos móveis de uma forma simples e livre.

Palavras chave: Android, HTTP Streaming, MPEG-DASH, MobileTV, Televisão.

Abstract

The evolution of technology carries out better ways of living. Therefore a necessity to adapt traditional habits to new and more modern ones emerges.

With this in mind, we came across this project developed in the scope of the Master in Electrical and Computer Engineering of the Faculty of Engineering of University of Porto. It aims to obtain a modern alternative to the conventional and sometimes outdated television.

To reach such solution it was developed a study about television and its used, or likely to be used, technologies for the signal diffusion. From this study it was concluded that the HTTP Streaming by MPEG-DASH would be the best option.

Thus it was developed a set of practical works focused on the development of an Android App, which allows interactions between the user as well as watching television through the mobile devices in a simple and free way.

Key words: Android, HTTP Streaming, MPEG-DASH, MobileTV, Television.

Agradecimentos

Sinto que não devo fazer uma lista de nomes e/ou instituições a quem devo um simples ou mais que merecido "Obrigado", sinto-o porque tenho a certeza que essa lista não iria conter todos os nomes meritórios de agradecimentos. O lapso a que o ser humano é associado iria certamente estar presente, o que me leva a não tornar esta secção de agradecimentos numa lista dos mesmos, porque não me sentiria bem quando tivesse de pedir desculpa a alguém por não "lhe ter agradecido".

Como tal deixo uma palavra de apreço a todas as pessoas que de uma forma ou de outra me auxiliaram durante este projeto. Além disso deixo uma enorme palavra de apreço a todas aquelas que durante todo este percurso académico, contribuíram de forma consciente ou não para fazer de mim uma pessoa diferente.

Obrigado pelas chatices, pelas horas perdidas noite dentro a discutir o mais insignificante dos problemas, pelos sorrisos, pelas gargalhadas e muito sinceramente pelas amarguras, que essas sim fazem muito mais por cada um de nós que as alegrias.

Lista de pessoas à parte, obrigado a todos aqueles que hoje posso dizer que são um pouco de mim e que certamente os vou levar, mais que não seja em memórias e pensamentos, quando deixar de frequentar esta instituição de ensino, a todos aqueles que neste momento "lhes esta a servir a carapuça", um sincero obrigado, a todos os outros a que estas palavras quase sem sentido não lhes diz nada, uma boa viagem para essa trajetória paralela que será as vossas vidas.

Cristhian Caldeira

“Quando tudo estiver a arder...
Aproveita e acende um cigarro.”

Paulus Phinus Esphomeatus

Conteúdo

| | | |
|----------|---|-----------|
| 1 | Introdução | 1 |
| 1.1 | Enquadramento | 1 |
| 1.2 | Objetivos | 1 |
| 1.3 | Motivação | 2 |
| 1.4 | Estrutura do Documento | 2 |
| 2 | Revisão Bibliográfica | 3 |
| 2.1 | Televisão Digital | 3 |
| 2.2 | Modulação | 4 |
| 2.3 | Compressão de Dados | 4 |
| 2.4 | Televisão Digital Interativa | 5 |
| 2.5 | Mobile TV | 6 |
| 2.5.1 | ISDB-T | 7 |
| 2.5.2 | MediaFLO | 7 |
| 2.5.3 | DMB | 7 |
| 2.5.4 | DVB-H | 8 |
| 2.6 | Streaming | 10 |
| 2.7 | HTTP (<i>Hypertext Transfer Protocol</i>) Streaming | 10 |
| 2.8 | HTTP Live Streaming | 11 |
| 2.8.1 | Arquitetura do HLS | 12 |
| 2.9 | MPEG-DASH | 13 |
| 2.9.1 | Media Presentation Description, MPD | 15 |
| 2.9.2 | Transmissão entre servidor HTTP e o cliente DASH | 15 |
| 2.10 | MobileTV em Portugal | 16 |
| 3 | Metodologia a Abordar e Ferramentas a Utilizar | 19 |
| 3.1 | Metodologia | 19 |
| 3.2 | Ferramentas a Utilizar | 19 |
| 3.2.1 | Android Studio | 20 |
| 3.2.2 | GPAC | 21 |
| 3.2.3 | MP4Box | 21 |
| 3.2.4 | ExoPlayer | 22 |
| 3.2.5 | HandBrake | 23 |
| 4 | Implementação Prática | 25 |
| 4.1 | Interatividade da Aplicação | 25 |
| 4.1.1 | Login com o Facebook | 25 |
| 4.2 | Reprodução de ficheiros DASH | 30 |

| | |
|---|-----------|
| 4.3 Criação e Alocação dos ficheiros DASH | 32 |
| 5 Conclusões e Trabalho Futuro | 45 |
| 5.1 Conclusão | 45 |
| 5.2 Trabalho Futuro | 46 |
| Referências | 47 |

Lista de Figuras

| | | |
|------|--|----|
| 2.1 | Resolução de ecrã para diferente formatos de TV [1]. | 3 |
| 2.2 | Conjunto de operações de uma STB num sistema TDI [2]. | 5 |
| 2.3 | ATSC, DVB e ISDB no mundo [3]. | 6 |
| 2.4 | Exemplo de uma distribuição media através de HTTP <i>Streaming</i> [4]. | 11 |
| 2.5 | Exemplo Arquitetura do HLS [5]. | 13 |
| 2.6 | Codificações com três resoluções diferentes para três segmentos [6]. | 14 |
| 2.7 | Alinhamento de vários segmentos com resoluções diferente [6]. | 14 |
| 2.8 | Demonstração do modelo de dados hierárquico do MPD [7]. | 15 |
| 2.9 | Comunicação servidor HTTP – Cliente DASH [7]. | 16 |
| 3.1 | Ambiente de desenvolvimento do Android Studio. | 20 |
| 3.2 | Logotipo GPAC Multimedia Open Source Project [8]. | 21 |
| 3.3 | Ambiente de funcionamento do Exoplayer – 2 Telas. | 22 |
| 3.4 | Ambiente de funcionamento do Handbrake. | 23 |
| 4.1 | Página inicial da aplicação com o botão de <i>login</i> através do facebook. | 29 |
| 4.2 | Pedido de dados para inicio de sessão através do facebook. | 29 |
| 4.3 | Classes e <i>layouts</i> incorporados do Projeto ExoPlayer. | 31 |
| 4.4 | Tela de inicio do ExoPlayer após <i>login</i> com o facebook. | 32 |
| 4.5 | Tela da aplicação após as alterações referidas ao ficheiro media.exolist.json . | 34 |
| 4.6 | Tentativa falhada de visualizar o conteúdo codificado anteriormente | 35 |
| 4.7 | Nova tela da aplicação após a segunda alteração do ficheiro media.exolist.json. | 37 |
| 4.8 | Tela da aplicação, sendo possível visualizar o vídeo mas não estando presente o áudio. | 37 |
| 4.9 | Nova tela da aplicação após a terceira alteração do ficheiro media.exolist.json. | 39 |
| 4.10 | Tela da aplicação, sendo possível visualizar o vídeo. | 40 |
| 4.11 | Tela da aplicação, sendo possível observar que a representação da componente de áudio esta presente. | 40 |
| 4.12 | Nova tela da aplicação após a última alteração ao ficheiro media.exolist.json. | 42 |
| 4.13 | Exemplo de uma reprodução com boa qualidade, da mudança entre resoluções e de uma reprodução com má qualidade. | 43 |

Lista de Tabelas

| | | |
|-----|--|----|
| 2.1 | Comparação dos serviços de <i>Mobile TV</i> vigentes 2011. | 17 |
|-----|--|----|

Abreviaturas e Símbolos

Abreviaturas

| | |
|-----------|---|
| AES | Advances Encryption Standard |
| AM | Amplitude Modulation |
| API | Aplication Programing Interface |
| ATSC | Advanced Television System Committee |
| AVC | Advanced Video Coding |
| CPU | Central Processing Unit |
| DMB | Digital Multimedia Broadcasting |
| DRM | Digital Rights Management |
| DVB | Digital Video Broadcasting |
| DVB-H | Digital Video Broadcasting - Handheld |
| DVB-T | Digital Video Broadcasting - Terrestrial |
| FLO | Forward Link Only |
| FM | Frequency Modulation |
| GPAC | Geelong Performing Arts Centre |
| HDTV | High - Definition Television |
| HEAAV | High - Efficiency Advanced Audio Coding |
| HLS | HTTP Live Streaming |
| HTTP | Hypertext Transfer Protocol |
| IDE | Integrated Development Enviroment |
| IP | Internet Protocol |
| ISDB | Integrated Services Digital Broadcasting |
| ISDB-T | Integrated Services Digital Broadcasting - Terrestrial |
| ISO | International Organization for Standardization |
| MPD | Media Presentation Description |
| MPE | Multiprotocol Encapsulation |
| MPE-FEC | Multiprotol Encapsulation - Forward Error Correction |
| MPEG | Moving Picture Experts Group |
| MPEG-DASH | Moving Picture Experts Group - Dynamic Adaptive Streaming over HTTP |
| MPEG-TS | Moving Picture Experts Group - Transport Stream |
| NHK | Nippon Hoso KyoKai |
| NTSC | National Television System Committe |
| OFDM | Orthognal Frequency - Division Multiplexing |
| PAL | Phase Alternative Line |
| QAM | Quadrature Amplitude Modulation |
| SDK | Software Development Kit |
| SECAM | Séquentiel Couleur à Mémoire |
| SFN | Single Frequency Network |
| STB | Set-Top-Box |
| TDI | Televisão Digital Interativa |
| TDMB | Terrestrial Digital Multimedia Broadcast |
| TMN | Telecomunicações Móveis Nacionais |
| TV | Television/Televisão |
| URI | Uniform Resource Identifier |

Símbolos

| | |
|-------|-----------------|
| bit/s | Bit por Segundo |
| Km | Quilometro |
| Hz | Hertz |
| € | Euro |
| ms | Milissegundo |

Capítulo 1

Introdução

1.1 Enquadramento

Com o avanço tecnológico vivido neste milénio cada vez mais temos o poder informativo na palma das mãos, e por consequência, velhos hábitos e máximas começam a cair por terra [9].

Neste caso e colocando o foco da nossa atenção na televisão convencional, conseguimos constatar que quase a totalidade da população acede com regularidade a conteúdos televisivos, no entanto, à medida que escalamos a faixa etária do utilizador, a fidelidade à caixinha mágica que adorna a maioria das salas de estar aumenta. Por outro lado, e este merece especial atenção, devido a tratar-se do futuro dos conteúdos televisivos, à medida que descemos a faixa etária do utilizador (com mais de 16 anos), o acesso a conteúdos televisivos, apesar de ser feito com regularidade tem cada vez menos o significado de "ver televisão". Desta forma a convergência entre TV e Internet de banda larga nos dispositivos móveis, possibilita o desenvolvimento de um grande número de novos serviços e produtos multimédia interativos que possibilitam novas experiências ao utilizador.

Os grandes avanços tecnológicos que se têm vindo a verificar nos dispositivos móveis, nomeadamente em relação às suas capacidades de reprodução de conteúdos audiovisuais e interface com o utilizador, abriu caminho para novas formas de distribuição de conteúdos televisivos. No entanto, tendo em conta as grandes diferenças que continuam a existir entre os terminais que tradicionalmente são utilizados para o consumo desse tipo de conteúdos e os terminais móveis, torna-se necessário conceber soluções que permitam gerar conteúdos adaptados a estas novas plataformas de consumo, quer em termos de características técnicas, quer mesmo em termos editoriais.

1.2 Objetivos

Dada a situação mencionada anteriormente, sobre a problemática desta dissertação, realçam-se algumas necessidades neste campo, tais como despertar o interesse do público

em adquirir os conteúdos em novas plataformas. Assim podemos definir como objetivos deste trabalho o estudo das tendências da indústria televisiva concebendo uma solução que permita disponibilizar conteúdos audiovisuais em dispositivos móveis. Esta solução passa pela conceção de uma aplicação que possibilite uma maior interatividade, aquando do consumo dos conteúdos televisivos, sendo para isto necessário conhecer os tipos de formatos e conteúdos bem como a interatividade que é desejada oferecer ao utilizador.

1.3 Motivação

"Logo que, numa inovação, nos mostram alguma coisa de antigo, ficamos sossegados."

Friedrich Nietzsche

Com este exemplo, podemos interpretar que serve de motivação para este trabalho e para desenvolver algo em redor desta problemática, o fato de conseguir trazer o sossego mencionada a um grande número de pessoas. Ao mesmo tempo que há a consciência que trabalhar em redor deste tema, depende de uma forma gigantesca da capacidade de aceitação por parte dos utilizadores.

Esta aceitação é, ou pode ser, influenciada por diversos fatores, sendo um deles o contexto económico do utilizador, desta forma e surgindo este desafio, nesta altura, em que cada vez menos se ouve falar em "crise", torna-se extremamente motivante trabalhar e desenvolver um projeto com capacidade para inovar, em diversas situações do dia a dia dos utilizadores e partindo de algo tão conhecido, que é a televisão.

1.4 Estrutura do Documento

No decorrer deste trabalho será feito um estudo sobre formas de Televisão existentes, técnicas e tecnologias para a difusão de sinais de televisão, caracterização da problemática bem como de metodologias e ferramentas utilizadas e apresentação dos trabalhos práticos elaborados e respetivos resultados e conclusões.

Capítulo 2

Revisão Bibliográfica

2.1 Televisão Digital

Remetendo a sua génese para a década de 70, a Televisão Digital surge no Japão quando a direção da rede pública de TV local, em conjunto com um grupo de cerca de cem estações comerciais, aprovaram que os cientistas do NHK (*Nippon Hoso Kyokai*) *Science and Technology Research Laboratories* desenvolvessem uma TV com qualidade de resolução superior aos formatos tradicionais de televisão até então conhecidos (NTSC (*National Television System Committee*), SECAM (*Séquentiel Couleur à Modulation*), PAL(*Phase Alternative Line*)) e que se destinava a receber o nome de HDTV (*High - Definition Television*).

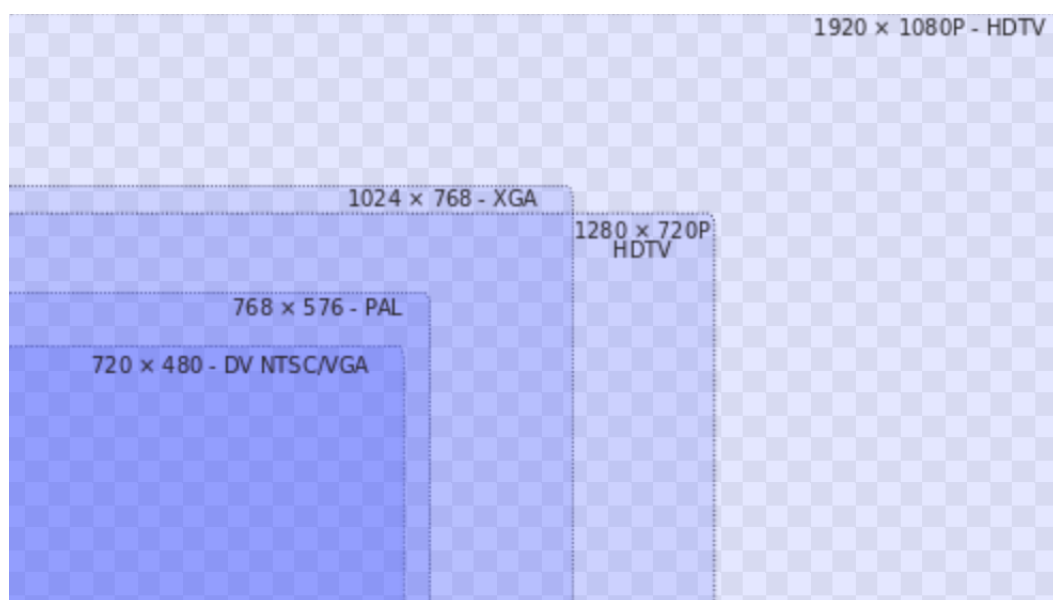


Figura 2.1: Resolução de ecrã para diferente formatos de TV [1].

Falando em Televisão Digital, é de extrema importância, para uma melhor compreensão, falar de dois pontos basilares para a sua existência, nomeadamente modulação e compressão de dados, pontos estes que seguidamente e de forma breve são clarificados.

2.2 Modulação

Modulação pode ser interpretado como o processo que faz variar a amplitude, frequência, comprimento e/ ou fase de uma onda de transporte, alterando uma das características (amplitude, fase ou frequência) de um sinal portador.

A maioria dos sinais, devido à forma como são fornecidos pelo transmissor não podem ser enviados diretamente através dos canais de transmissão. É desta limitação que acaba por emergir a importância da modulação, surgindo então a necessidade de modificar os sinais a transmitir, recorrendo para tal a uma onda eletromagnética portadora que dada a sua natureza reúne um leque de características bastante favoráveis aos meios de transmissão.

Em suma o processo de modulação é nada mais, nada menos, que alterar um parâmetro da onda portadora em função da variação do sinal modulante, que por sua vez é a informação que se pretende transmitir [10] [11].

Dependendo do parâmetro sobre o qual se atua, existem diversos tipos de modulação, mencionando que as mais usuais são sem dúvida a Modulação em Amplitude - AM e a Modulação em frequência - FM.

2.3 Compressão de Dados

Numa forma simples, compressão de dados é o processo de reduzir o espaço ocupado por dados. Muitas vezes destina-se apenas a extrair informação redundante, informação esta que pode ser removida sem alterar a forma ou o conteúdo dos dados originais. É evidente que tal processo tem como objetivo a diminuição dos custos (também computacionais) de armazenamento, permitindo uma diminuição da quantidade de dados que precisam de ser movidos em operações. É importante ter atenção quando deve acontecer este processo, pois a compressão e descompressão sistemática e repetitiva de dados ao longo de processos pode causar um notável *overhead* do CPU (*Central Processing Unit*), aumentando consideravelmente o tempo de resposta de certas operações, denotando-se desta forma a importância de manter os dados de forma comprimida o maior tempo possível, procedendo-se a descompressão apenas quando for realmente necessário.

A classificação de compressão de dados pode ser feita de diversas formas, sendo a mais usual pela ocorrência ou não de perdas de informação durante o processo. Assim, e de forma sucinta estamos perante uma compressão sem perdas, se após a compressão e descompressão, os dados obtidos forem idênticos aos dados originais, sendo primordial a sua utilização em situações onde uma perda pequena de dados possa comprometer a utilização dos mesmos. Recorrendo a um exemplo bastante trivial, no caso de compressão de um texto, não poderá ser utilizada uma compressão com perdas, dado que um texto com letras trocadas ou sem algumas delas tornar-se-á muito facilmente inútil, dado que a sua compreensão e consequentemente utilização fica desde logo comprometida.

De forma bastante similar estamos perante uma compressão com perdas, quando os dados após comprimidos apresentam perdas de informação pouco significativas face aos dados originais, sendo a sua utilização possível em situações em que esta perda de informação não comprometa a utilização dos mesmos. Por exemplo quando estamos perante uma compressão de uma imagem e as perdas de informação resultantes da compressão não são perceptíveis a olho nu [10] [12].

2.4 Televisão Digital Interativa

Sendo a televisão digital interativa uma consequência da televisão digital e um sistema que se projeta para possibilitar interatividade com o utilizador, esta tem a obrigação de conseguir conjugar diferentes tecnologias de *hardware* e *software* para melhor implementar as suas atrativas funcionalidades. Assim o sinal eletromagnético que transporta segmentos elementares de informação áudio, vídeo, dados e aplicações pode ser transmitido à *set-top-box*, que por sua vez os irá processar e apresentar ao utilizador.



Figura 2.2: Conjunto de operações de uma STB num sistema TDI [2].

Devido à diversidade de implementações possíveis em sistemas de televisão interativa, surgiram três padrões, reconhecidos e utilizados mundialmente (ATSC – *Advanced Television System Committee*, DVB – *Digital Video Broadcasting*, ISDB – *Integrated Services Digital Broadcasting*), que de uma forma bastante concisa diferem uns dos outros no que diz respeito à modulação do sinal de difusão, transporte de informação elementar de áudio, vídeo e aplicações, bem como na qualidade de áudio e vídeo. Posteriormente é possível abordar mais profundamente estes e outros padrões existentes [10] [13].

- MediaFLO da norte americana Qualcomm;
- TDMB (*Terrestrial Digital Multimedia Broadcasting*) de origem coreana;
- DVB-H (*Digital Video Broadcasting – Handheld*) financiado pela comunidade europeia.

2.5.1 ISDB-T

Começou a ser desenvolvido nos anos 70 pelo laboratório de pesquisa TV NHK e, devido à sua resposta à necessidade de mobilidade e portabilidade, é visto como o *standard* mais flexível comparativamente com todos os outros, dado à sua grande versatilidade ao prever serviços de transmissão para dispositivos fixos ou móveis através do mesmo sinal.

Usa a norma H.262/MPEG-2 Parte 2 para a compressão de vídeo e dispõe de uma taxa de apresentação de 30 *frames* por segundo e de 15 *frames* para dispositivos móveis. Importa também realçar que para além da transmissão de sinais de televisão digital este *standard* pode ser utilizado para outras finalidades, destacando-se a transmissão de dados e receção por meio de computador ou servidor doméstico [15].

2.5.2 MediaFLO

Media Forward Link Only, como o próprio nome indica só envia dados aos dispositivos móveis, não recebendo nenhum retorno de dados destes.

Esta tecnologia para a transmissão de áudio, vídeo e dados para dispositivos móveis foi desenvolvida pela Qualcomm devido principalmente aos problemas existentes no que diz respeito ao consumo de baterias, na implementação da televisão analógica tradicional e sinais terrestres de televisão digital.

Os dados transmitidos pela MediaFLO são também bastante diversificados, incluindo fluxos em tempo real de áudio e vídeo, dados de aplicações de *datacast* do *Internet Protocol*, como resultados desportivos, informações meteorológicas bem como cotações de mercados [16].

2.5.3 DMB

É um sistema de transmissão de rádio digital, desenvolvido na Coreia do Sul, que envia sinais de rádio, TV e *datacasting* para dispositivos móveis.

Esta destina-se à banda III (VHF) e a banda L (HHF) das frequências de rádio terrestres, o que se revelou ser um ponto negativo na sua implementação prática dado que tanto os Estados Unidos como o Canadá ainda terem a primeira banda alocada para transmissão de televisão e a segunda banda estar reservada por parte dos Estados Unidos para uso militar. No entanto existem pontos positivos da sua implementação prática, sendo possível após ensaio experimental mencionar que o sistema DMB funciona sem problemas em dispositivos que se estejam a mover a uma velocidade superior a 120 km/hora, possibilitando assim tanto a visualização de conteúdos para utilizadores que recorrem aos convencionais terminais móveis

em carros, comboios e outros meios de transporte, como também permite a implementação de raiz de tecnologia nos próprios transportes, bem como em situações de utilização dentro de túneis, foi possível verificar que o serviço de TV e rádio continuam a funcionar, existindo por vezes algumas paragens que nunca excederam os 2 segundos até serem retomadas.

2.5.4 DVB-H

O grande sucesso do DVB-T (*Digital Video Broadcasting - Terrestrial*), os seus benefícios e o potencial da transmissão digital, atraiu a atenção da indústria da área dos dispositivos móveis. Tendo como objetivo responder a este interesse, a organização DVB Project desenvolveu um novo *standard* de transmissão digital destinado às comunicações móveis, o DVB-H, que é uma especificação técnica para importar serviços de *broadcast* para telemóveis, e foi oficialmente implementada pela União Europeia como a tecnologia para radiodifusão móvel terrestre em Março de 2008.

Esta tecnologia assenta em quatro princípios basilares:

- Garantir uma "qualidade aceitável" na visualização dos conteúdos;
- Cobertura geográfica;
- Possibilitar acesso ao serviço enquanto o utilizador se desloca a velocidades consideráveis, por exemplo, enquanto se desloca num veículo;
- Compatibilidade com a televisão digital terrestre existente, de forma a permitir partilha de equipamentos de rede e de transmissão.

A tecnologia DVB-H é um superconjunto do bem sucedido sistema DVB-T para televisão digital terrestre, com recursos adicionais para atender aos requisitos específicos dos receptores portáteis alimentados por baterias, que se consideram ser pequenos, leves e portáteis, assentando a sua arquitetura na camada física do DVB-T, apresentando no entanto a nível da camada de ligação alterações com o objetivo de respeitar os requisitos mencionados em cima.

Antes de uma abordagem mais técnica desta tecnologia é relevante mencionar que a mesma é capaz de proporcionar um canal de *downstream* em altas taxas, que pode ser usado como autónomo ou como reforço de redes de telecomunicações móveis, que muitos terminais *handheld* típicos são capazes de aceder de qualquer maneira.

No que se refere a transmissão, o DVB-H utiliza o sistema IP (*Internet Protocol*), possibilitando interligações com outras redes baseadas neste protocolo. Por forma a otimizar a compatibilidade da utilização de redes de transporte, o DVB-H recorre ao protocolo MPE (*Multiprotocol Encapsulation*), que permite o encapsulamento de pacotes IP em pacotes TS, permitindo assim o uso das redes de transporte do DVB-T que por sua vez usa o MPEG-TS (*Moving Picture Experts Group - Transport Stream*).

A nível da camada de ligação, o sistema DVB-H caracteriza-se pelas importantes funcionalidades de *time-slicing* e de MPE-FEC (*Multiprotocol Encapsulation - Forward Error Correction*).

O *time-slicing* é usado com o intuito de minimizar o consumo de energia por parte dos terminais móveis, pois, assim os datagramas IP são transmitidos em "rajadas" de dados em curtos intervalos de tempo, podendo cada conjunto destes conter até 2 megabits de dados, o que por sua vez possibilita que o *front-end* do recetor tenha apenas de se ligar durante o intervalo de tempo em que a "erupção" de dados de um serviço está no ar. Durante este curto espaço de tempo dá-se a receção de uma elevada taxa de dados, que pode ser armazenada num *buffer*, podendo este armazenar os dados das aplicações descarregados ou transmitir fluxos em tempo real.

De forma a garantir alguma resistência a erros o DVB-H usa o MPE-FEC.

A nível da camada física são introduzidas quatro extensões:

- Introdução de sinalização DVB-H nos bits TPS, que permite indicar a presença de serviços DVB-H, bem como o uso do mecanismo opcional MPE-FEC;
- O modo de transmissão OFDM (*Orthogonal Frequency - Division Multiplexing*) 4K, que facilita o desenho de redes SFN (*Single Frequency Network*), apropriadas para a receção de sinais de dispositivos móveis;
- Possibilidade de utilizar um método de entrelaçamento não nativo do modo de transmissão, que melhora significativamente a robustez a efeitos de *fading* seletivo de frequências,
- Além de suportar as larguras de banda oferecidas pelo sistema DVB-T, existe a possibilidade de o DVB-H usar um canal com largura de banda de 5 MHz.

2.5.4.1 Codificação de Áudio e Vídeo (DVB-H)

O sistema DVB-H não usa o *standard* MPEG-2 como método de codificação, este facto deve-se à natureza dos dispositivos móveis, nomeadamente, às suas pequenas dimensões, o que exige uma codificação de vídeo mais eficiente que a norma MPEG-2, que suporte os baixos débitos e as baixas resoluções. Desta forma é utilizada a norma H.264/AVC para a codificação de vídeo.

Na parte da codificação de áudio é então utilizado o HE AAC v2 definido na norma MPEG-4, pois este possibilita níveis de *bit rate* muito baixos, mantendo uma boa qualidade de áudio.

2.5.4.2 Modo de transmissão (DVB-H)

Como já foi anteriormente mencionado o modo de transmissão no sistema DVB-H é uma extensão da camada física do sistema DVB-T. O OFDM pode ter três modos:

- 2K, bastante utilizado para redes SFN de pequenas dimensões, dado que oferece um maior intervalo entre as frequências portadoras, apresentando também uma duração de símbolo muito baixa e, por isso, um intervalo de guarda pequeno, o que por sua vez aumenta consideravelmente a probabilidade de ocorrência de interferências na receção.
- 8K, modo que beneficia o recetor pois facilita-lhe a interpolação de cada sinal, sendo mais adequado para redes SFN de larga escala. No entanto não se apresenta como a melhor opção para dispositivos móveis com apenas uma antena, dados apresentar fracos resultados na receção.
- 4K, surge assim como a melhor opção para o DVB-H, possibilitando a cada sub-portadora da modulação OFDM usar uma modulação 16-QAM ou 64-QAM, sendo a primeira alternativa mais robusta embora apresente um *bit rate* mais baixo.

Para concluir esta abordagem sobre transmissão DVB-H é importante mencionar o mecanismo de *in-depth interleaved*, que aumenta a flexibilidade da interpolação de símbolo, permitindo que um sinal no modo 4K tenha benefícios da memória de interpolação de símbolo do modo 8K. Assim para o modo 4K é possível dobrar a interpolação de símbolo, o que se resume numa melhoria bastante significativa na receção do sinal, aumentando a proteção contra ruído.

2.6 *Streaming*

É uma forma de distribuição de conteúdos digitais, onde contrariamente ao *download* de ficheiros, as informações não são armazenadas nos dispositivos dos utilizadores. Pode mencionar-se que esta forma de distribuição é vantajosa porque:

- Não é ocupado espaço no dispositivo do utilizador para a reprodução, exceto para o arquivo temporário na *cache* do sistema.
- O fluxo de dados é recebido e os conteúdos são reproduzidos à medida que chegam ao utilizador, dependendo da largura de banda.
- Possibilita que o utilizador assista a conteúdos protegidos por direitos de autor, sem a violação desses mesmos direitos, tal como acontece no caso do rádio e da televisão.
- Oportunidade de transmissão em diversas plataformas, *Multicast* ou *Broadcast*.

2.7 HTTP (*Hypertext Transfer Protocol*) *Streaming*

Apresentando-se como uma alternativa para o *MobileTV*, são inúmeras as características associadas ao HTTP *Streaming*

Inicialmente é de realçar toda a estrutura atual da Internet, que evoluiu e continua a evoluir de forma a oferecer cada vez mais e um melhor suporte ao HTTP. Para além de que este protocolo é *“firewall friendly”*, dado que quase todas as *firewalls* estão configuradas para suportar as suas conexões.

Em segundo lugar, com o fluxo HTTP, o cliente gera a transmissão sem necessitar de manter um estado de sessão no servidor, desta forma um grande número de clientes de transmissão não é sinónimo de custos adicionais, além dos do normal da Web do HTTP.

Desta forma o *streaming* HTTP tornou-se uma abordagem popular em implementações comerciais, destacando-se entre outras implementações o [7]:

- HTTP *Live Streaming* da Apple;
- *Smooth Streaming* da Microsoft;
- *Streaming* dinâmico da Adobe;
- MPEG-DASH (*Moving Picture Experts Group - Dynamic Adaptive Streaming over HTTP*).

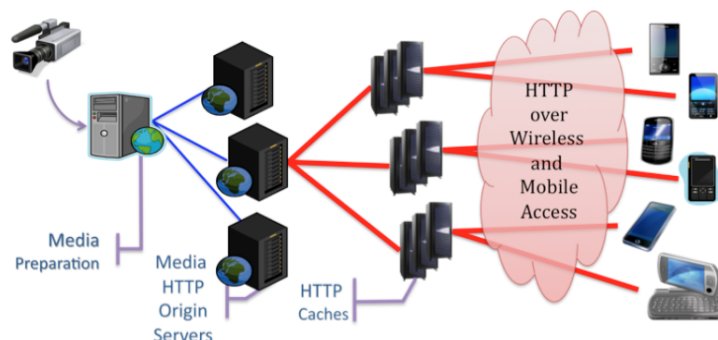


Figura 2.4: Exemplo de uma distribuição media através de HTTP *Streaming* [4].

Por serem bastante semelhantes e uma das hipóteses a adotar na implementação prática, será feita uma abordagem quer sobre o HLS (HTTP Live Streaming) quer sobre o MPEG-DASH.

2.8 HTTP Live Streaming

Este protocolo de transmissão baseado em HTTP, criado pela Apple Inc, é bastante semelhante ao MPEG-DASH, como será possível denotar na secção seguinte, na medida em que o conteúdo *media* a ser transmitido é dividido em uma sequência de pequenos segmentos de arquivos, onde cada segmento contém a informação referente a um curto intervalo de tempo da reprodução de um conteúdo muito maior.

São disponibilizados segmentos alternativos codificados a diferentes taxas de bits que abrangem intervalos curtos e alinhados de tempo. Assim, enquanto um conteúdo referente a um segmento esta a ser reproduzido, o cliente seleciona automaticamente das alternativas possíveis, o próximo segmento a ser descarregado e reproduzido conforme as condições da rede no momento, ou seja, a qualidade do conteúdo a ser reproduzido é adaptado conforme o congestionamento da rede, em curtos intervalos de tempo.

Existem pontos chaves no sucesso inerente ao HLS:

- Como já foi anteriormente referido, dado todas as solicitações usarem apenas transações HTTP, este protocolo garante a passagem por todas as *firewalls* que permitam o tráfego HTTP ;
- A possibilidade de o conteúdo estar alocado e ser disponibilizado por servidores HTTP convencionais, que são e estão amplamente disponíveis atualmente;
- Presença de um método de codificação avançado e seguro, método de Rijndael (AES – Advances Encryption Standard).

2.8.1 Arquitetura do HLS

O HTTP *Live Streaming* é constituído por 3 elementos principais:

- Servidor: que requer um codificador e um segmentador. É responsável por receber fluxos de entrada *media* codifica-los e encapsula-los num formato adequado para a entrega;
 - Codificador : recebe um sinal em tempo real, codifica o conteúdo e trata do encapsulamento (fluxos de transporte MPEG-2);
 - Segmentador : divide o arquivo MPEG-2 TS em fragmentos de igual comprimento, mantidos como arquivos .ts.
- Distribuidor: constituído pelos servidores Web comuns, responsável por aceitar os pedidos do cliente e entregar toda a *media* preparada pelo servidor, bem como todos os recursos necessários para o *streaming*;
- Cliente : é responsável por determinar todo o conteúdo a solicitar, descarregar esses conteúdos e seguidamente reconstitui-los de forma a que o utilizador os possa ver num fluxo contínuo. Para tal, inicialmente é descarregado o arquivo de índice, através de um URL e seguidamente os inúmeros arquivos de *media* disponíveis [5].

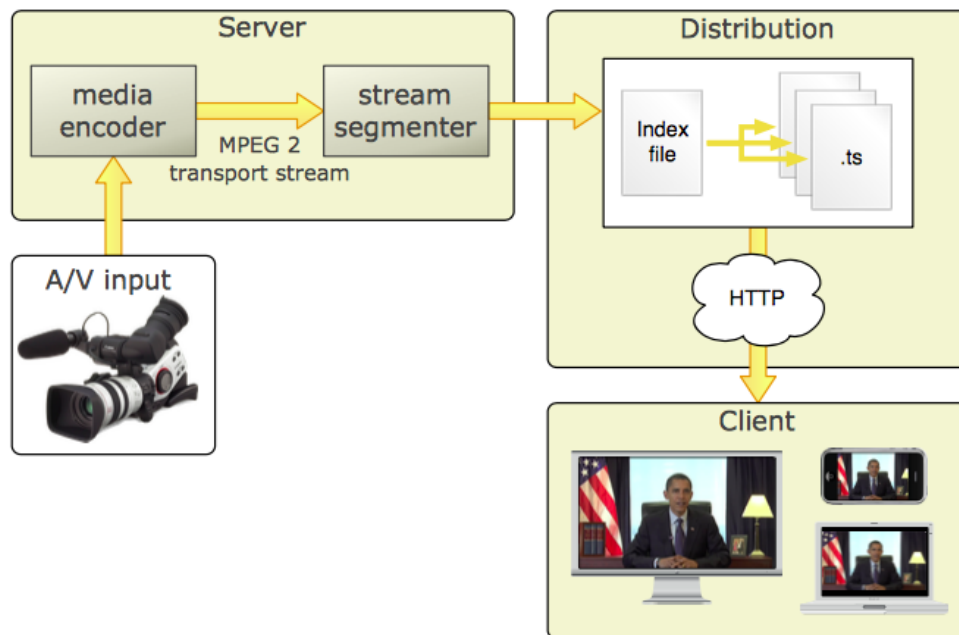


Figura 2.5: Exemplo Arquitetura do HLS [5].

2.9 MPEG-DASH

Em Abril de 2009 o MPEG após observar as perspectivas de mercado e os pedidos da indústria emite uma proposta para um *standard* de transmissão de HTTP.

Durante os dois anos seguintes o MPEG desenvolve trabalhos juntamente com a participação de vários especialistas, dos quais resultou o *standard* MPEG Dynamic Adaptive Streaming over HTTP.

De forma bastante semelhante ao HLS como já foi anteriormente referido, o MPEG-DASH é uma tecnologia de *streaming* de fluxo de bits adaptável, onde um arquivo multimédia é particionado em um ou mais segmentos e entregue a uma cliente usando HTTP.

O sistema DASH funciona basicamente da seguinte forma:

- Codificador de vídeo:

Inicialmente o ficheiro de vídeo é codificado em diferentes versões, contando cada uma delas com uma taxa e/ou resolução diferentes. As representações de um vídeo têm o mesmo conteúdo, mas diferem entre si na qualidade do mesmo, sendo cada um destes ainda divididos no tempo em segmentos iguais.

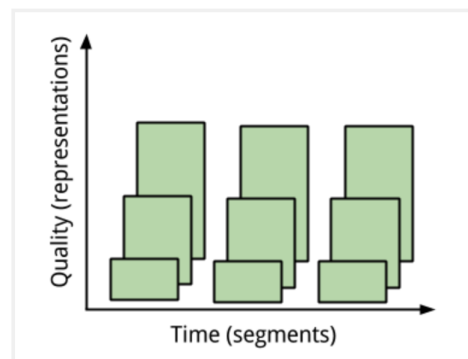


Figura 2.6: Codificações com três resoluções diferentes para três segmentos [6].

- *Manifest:*

O arquivo de manifesto no MPEG-DASH, é chamado de *Media Presentation Description*, MPD. Este é nada mais nada menos que um arquivo XML, que tem como função identificar as várias representações, fornecer informações sobre a resolução do vídeo e sobre a taxa de reprodução, para cada segmento, bem como indicar a localização dos mesmos em cada representação, sendo este arquivo armazenado no servidor juntamente com os arquivos de vídeo.

- Entrega de vídeo:

Após os passos mencionados anteriormente estarem concluídos, o cliente começa por solicitar o MPD. Após obter as informações contidas neste, é possível decidir qual a representação pretendida para o primeiro segmento. Seguidamente, o arquivo pretendido é descarregado, sendo usado para isso o URL contido no MPD, podendo assim dar-se a reprodução do mesmo.

Por forma a não introduzir paragem ou interrupções no *streaming*, de cada vez que um arquivo é descarregado, o cliente volta a analisar as condições atuais e toma uma nova decisão sobre a representação seguinte para obter o segmento que se segue.

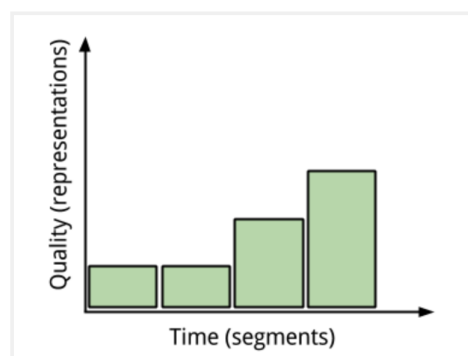


Figura 2.7: Alinhamento de vários segmentos com resoluções diferente [6].

2.9.1 Media Presentation Description, MPD

É importante ter a noção que o HTTP *streaming* exige varias alternativas de taxas de bits do conteúdo multimédia que devem estar disponíveis no servidor. Além de que o conteúdo multimédia é constituído por vários componente de *media*, tendo cada um deles características próprias e bastante diferentes.

No MPEG-DASH, as informações referentes a essas características são descritas por um documento XML denominado MPD.

O MPD consiste em um ou vários períodos, onde um período é um intervalo do programa ao longo do tempo, tendo por sua vez cada período uma hora e duração de início e consiste em uma ou múltiplas adaptações. Cada conjunto de adaptações normalmente inclui múltiplas representações, dado que um conjunto de adaptações pode conter as diferentes taxas de bits da componente de vídeo e outro conjunto de adaptações pode por sua vez conter as diferentes taxas de bits da componente de áudio, do mesmo conteúdo multimédia.

Uma representação é uma alternativa codificada da mesma componente multimédia, variando as representações umas das outras por diferentes taxas de bits e resolução entre outras características. Cada representação consiste em um ou múltiplos segmentos, que por sua vez são blocos de fluxos *media* em sequência temporal, possuindo cada um deles o respetivo URL que indica a sua localização no servidor [7].

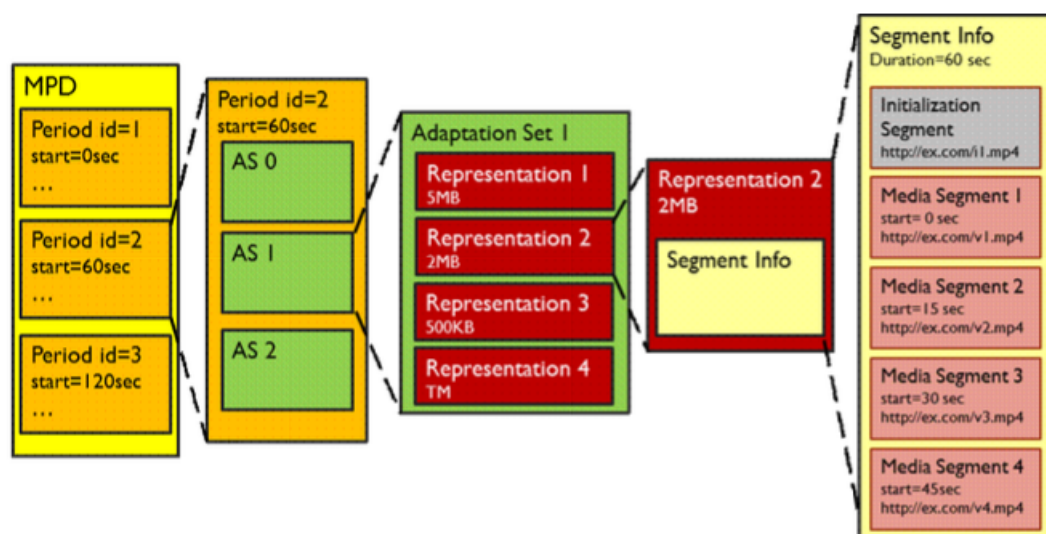


Figura 2.8: Demonstração do modelo de dados hierárquico do MPD [7].

2.9.2 Transmissão entre servidor HTTP e o cliente DASH

Como já foi possível abordar, o conteúdo existe no servidor em duas partes:

- MPD, que descreve um manifesto do conteúdo disponível, as suas alternativas, respetivos endereços URL e outras características;

- Segmentos, que contêm fluxos de bits multimédia reais, em arquivos únicos ou múltiplos.

O MPD é por norma entregue ao cliente por HTTP, podendo no entanto ser por e-mail ou outro método de transporte. Ao analisar o MPD, o cliente adquire conhecimentos sobre o *timing* do programa, a disponibilidade de conteúdos, as resoluções, as larguras de banda máxima e mínima bem como da existência de várias alternativas codificadas de componentes multimédia.

Através de DRM, *Digital Rights Management*, o cliente DASH seleciona a codificação apropriada e inicia a transmissão do conteúdo através da pesquisa dos segmentos, recorrendo para isso a solicitações HTTP GET.

Após o armazenamento num *buffer* apropriado para permitir as variações da taxa de transferência da rede, o cliente continua a pesquisa dos segmentos seguintes e também monitoriza as flutuações da largura de banda da rede.

É de salientar que a especificação MPEG-DASH apenas define o MPD e os formatos de segmento. A entrega do MPD, os formatos de codificação de *media* que contêm os segmentos, bem como o comportamento do cliente para a sua procura heurística de adaptação e reprodução do conteúdo, estão fora da vigência do MPEG-DASH.[7]

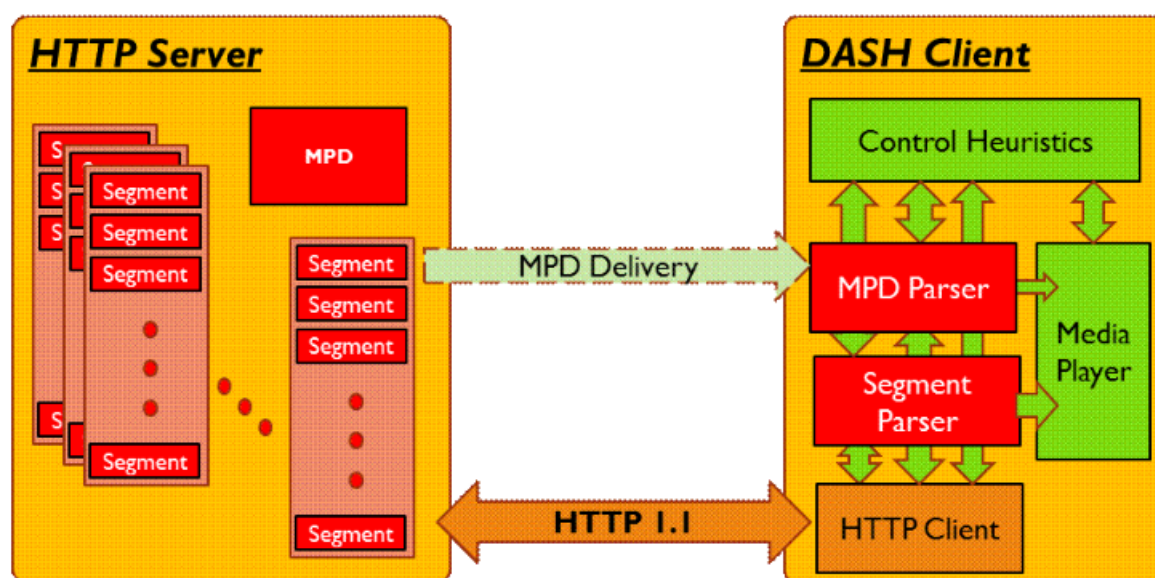


Figura 2.9: Comunicação servidor HTTP – Cliente DASH [7].

2.10 MobileTV em Portugal

O serviço de *MobileTV* entra no mercado português em 2006 através da Vodafone, que na altura oferecia dois tipos de subscrições, uma mensal (7,5€/mês), e outra de 24 horas (1,5€/dia), contendo uma lista não muito vasta de canais disponíveis, cerca de 14 canais.

Passados apenas três meses a TMN (atualmente MEO) disponibiliza aos seus clientes o serviço de *MobileTV*, contando com 21 canais na sua listagem.

Em 2011, o serviço já era assegurado pelas três principais operadoras nacionais (TMN, Vodafone e Optimus), sendo que de operadora para operadora, existia variações de quantidades de canais disponíveis, bem como do preço por subscrição, como se ilustra na tabela seguinte [17].

| Operadora | Nº Canais | Subscrição Semanal | Subscrição Mensal |
|-----------|-----------|--------------------|-------------------|
| TMN (MEO) | 40 | 2,14€ | 8,06€ |
| Vodafone | 32 | 2,01€ | 7,57€ |
| Optimus | 27 | 2,04€ | 7,69€ |

Tabela 2.1: Comparação dos serviços de *Mobile TV* vigentes 2011.

No entanto o sucesso do serviço de *MobileTV* em Portugal ficou muito à quem das expectativas. Estima-se que tal insucesso se deveu à pouca adesão aos pacotes de dados oferecidos pelas operadoras móveis, à falta de equipamento com características compatíveis com o serviço por parte dos utilizadores, bem como aos custos associados à subscrição do serviço. Em suma é possível de certa forma afirmar que o insucesso do *MobileTV* em Portugal se deveu ao contexto económico e ao fraco poder de compra nacional.

Capítulo 3

Metodologia a Abordar e Ferramentas a Utilizar

3.1 Metodologia

Após o estudo feito na secção anterior, foi tomado como ponto de partida para a implementação prática, a concessão de um serviço de *MobileTV*, baseado em HTTP *Streaming*, mais especificamente usando MPEG-DASH. Esta decisão deve-se às suas características, já visadas anteriormente, que permitem satisfazer de uma forma mais completa os objetivos traçados para este trabalho em redor da problemática que é o serviço de *MobileTV*, objetivos estes que consistem em conceber uma solução o mais universal possível, que possibilite interceção com os utilizadores, que seja facilmente aplicada dadas as condições atuais dos meios de comunicação e além disso que seja de acesso rápido e totalmente gratuito.

No capítulo seguinte é abordado o trabalho prático desenvolvido, com o objetivo de implementar a solução anteriormente mencionada, para tal, este trabalho foi desenvolvido apontando para uma alternativa baseada num dos sistemas operativos para dispositivos móveis mais utilizados mundialmente, o Android.

Dado que existe o interesse de proporcionar diversos níveis de interação com o utilizador, houve o recurso à atual rede social mais utilizado em todo o mundo, para proporcionar essa interação, o *Facebook*.

E finalmente por forma a ser possível a visualização de conteúdos, foi implementado um reprodutor de media, que é um projeto já existente da Google e *open source*, de forma a permitir a reprodução dos ficheiros, o ExoPlayer.

3.2 Ferramentas a Utilizar

No capítulo seguinte é descrito todos os passos para o desenvolvimento de uma aplicação, como já foi anteriormente mencionado, para o sistema operativo Android, portanto a principal ferramenta a ser mencionada neste capítulo é sem duvida o IDE utilizado para o

desenvolvimento da mesma, que é o Android Studio. Outras ferramentas ou projetos merecem também ser mencionados, como é o caso do GPAC (Geelong Performing Arts Centre), o Mp4Box, HandBrake e o ExoPlayer.

3.2.1 Android Studio

É um ambiente de desenvolvimento, que foi lançado pela Google, especificamente para o sistema operativo Android. A escolha deste IDE (*Integrated Development Enviroment*) deve-se essencialmente a 3 características:

- A facilidade de quando é criado um projeto, toda a estrutura do projeto aparecer com praticamente todos os ficheiros dentro do diretório SDK (*Software Development Kit*), o que oferece uma maior flexibilidade ao processo de construção.
- Permite ver qualquer alteração visual que for feita na aplicação que está a ser desenvolvida em tempo real;
- Utilização de um sistema de "arrastar e largar" para mover os componentes em toda a interface do utilizador.

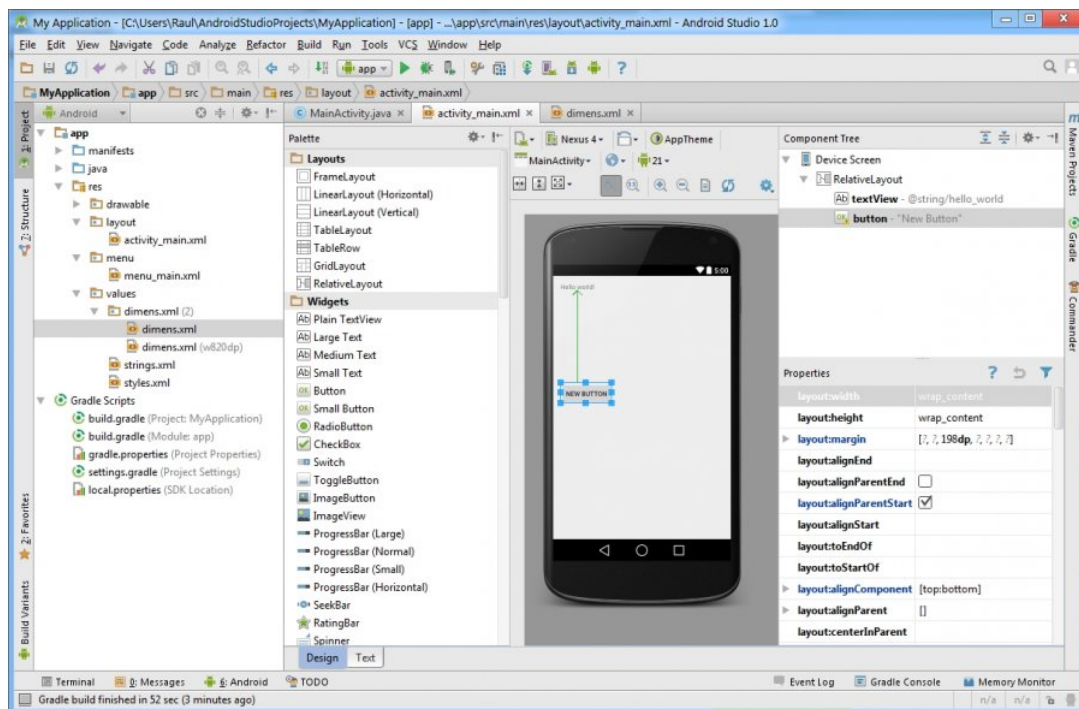


Figura 3.1: Ambiente de desenvolvimento do Android Studio.

3.2.2 GPAC

É um projeto *open source* que se destina a uma ampla audiência, que vai desde estudantes ou criadores de conteúdo que pretendam experimentar os novos padrões para tecnologias interativas, ou que pretendam converter arquivos para dispositivos móveis, para programadores que necessitem de reprodutores e/ou servidores para aplicações de transmissão multimédia. Este projeto abrange diferentes aspetos multimédia, tendo como foco as tecnologias de apresentação (animação e interatividade) e os formatos multimédia, como por exemplo o MP4.

O GPAC fornece os seguintes três conjuntos de ferramentas:

- Um reprodutor, chamado Osmo4/MP4Client;
- Um pacote multimédia, chamado MP4Box;
- Várias ferramentas de servidor incluídas nas aplicações MP4Box e MP42TS [8].



Figura 3.2: Logotipo GPAC Multimedia Open Source Project [8].

3.2.3 MP4Box

É um pacote multimédia disponibilizado no GPAC, que representa uma grande versatilidade no que diz respeito a manipulação sobre arquivos multimédia, destacando-se algumas das suas utilizações:

- Para manipular arquivos ISO (*International Organization for Standardization*): adicionar, remover, multiplexar dados de áudio, vídeo e texto de diferentes fontes e em diferentes formatos;
- Para codificar/descodificar linguagens de apresentação a partir de formatos binários;
- Para a preparação de conteúdo *HTTP Adaptive Streaming*.

Sendo um exemplo da sua utilização a seguinte linha de código:

```
MP4Box -dash 1000 - frag 1000 - rap -segment_ -out teste.mpd teste.mp4
```

3.2.4 ExoPlayer

O ExoPlayer é um reproduztor multimédia disponibilizado pela Google, como um projeto *open source*. A sua escolha para a implementação prática neste trabalho deve-se essencialmente ao facto de ser uma alternativa à API (Aplication Programing Interface) do MediaPlayer do Android para reproduzir áudio e vídeo, tanto ficheiros locais como derivados da Internet, suportando recursos atualmente não suportados pelo MediaPlayer do Android, incluindo as representações adaptativas DASH. Sendo também um forte ponto a favor da sua implementação neste trabalho a sua fácil personalização e expansão.

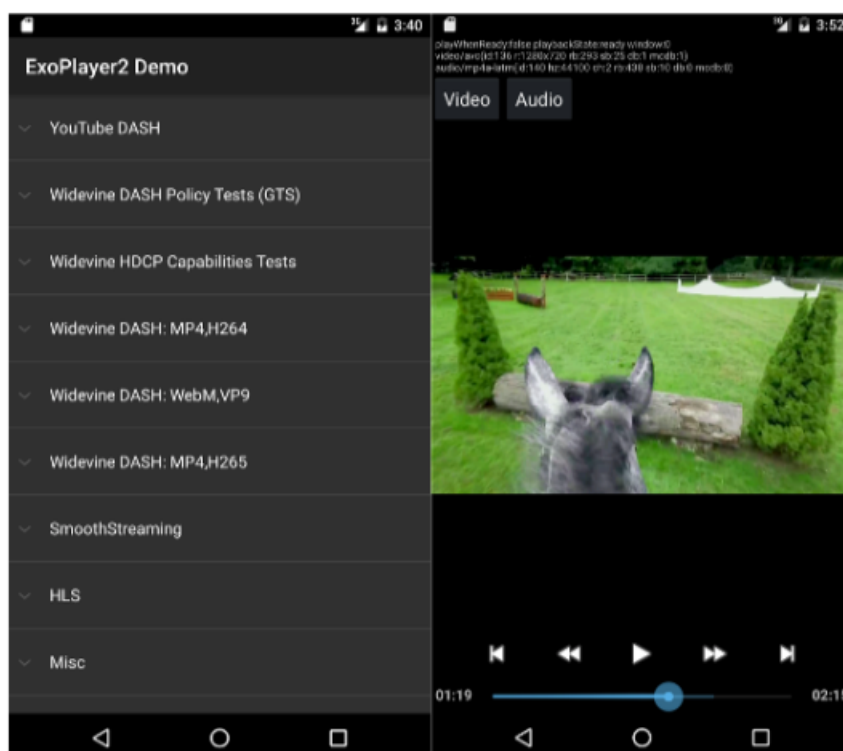


Figura 3.3: Ambiente de funcionamento do Exoplayer - 2 Telas.

3.2.5 HandBrake

É mais uma ferramenta *open source* utilizada para converter vídeos (de quase todos os formatos) para uma seleção de *codecs* modernos e amplamente suportados. Possibilitando a criação de um ficheiro vídeo com um dada resolução, a partir de um ficheiro original com uma resolução completamente diferente.

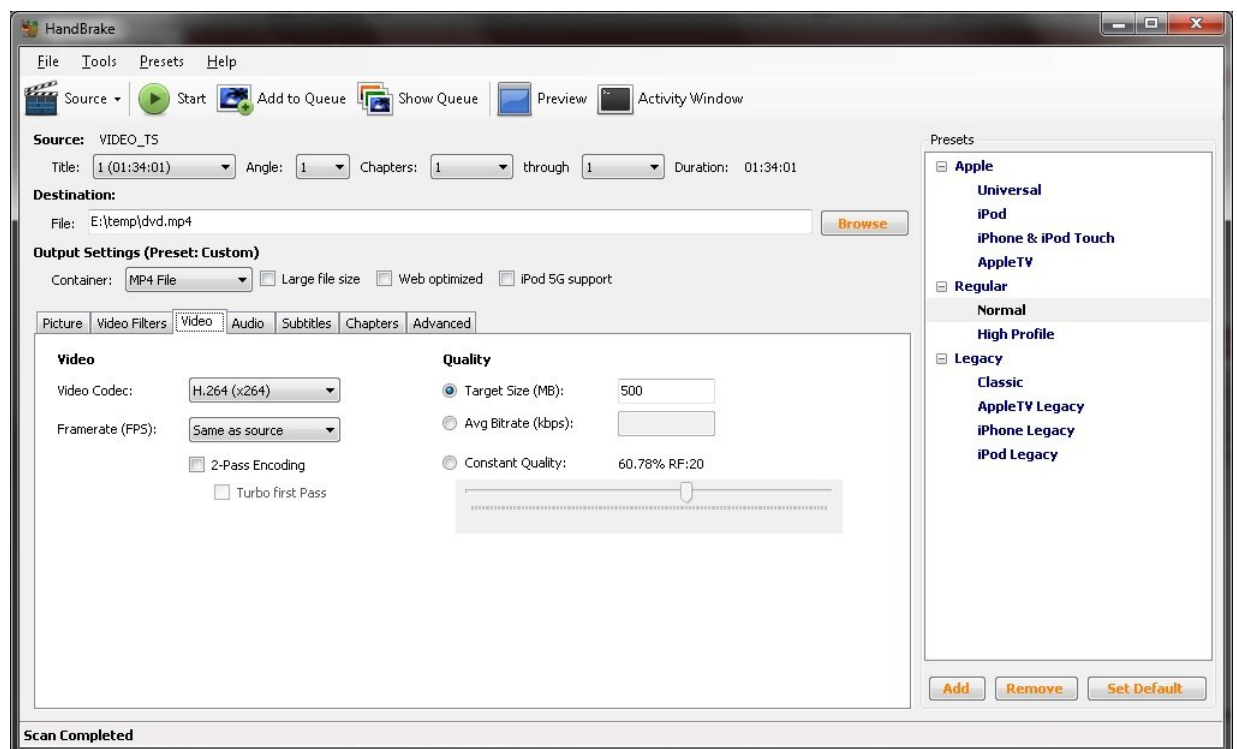


Figura 3.4: Ambiente de funcionamento do Handbrake.

Capítulo 4

Implementação Prática

Neste Capítulo, é mostrado o trabalho prático elaborado e a forma como o mesmo foi feito, visando a criação de uma aplicação Android, capaz de reproduzir ficheiros MPEG-DASH, de forma eficiente e gratuita. Passo a passo, é possível ver todas as etapas da construção desta aplicação, nomeadamente:

- Interatividade da aplicação;
- Reprodução de ficheiros DASH;
- Criação e alocação dos ficheiros DASH.

4.1 Interatividade da Aplicação

4.1.1 Login com o Facebook

Assim, e com o intuito de estabelecer conexão entre este trabalho prático e as redes sociais, foi seguido para a implementação do *login* pelo facebook, toda a informação e documentação existente na página disponibilizada pelo facebook para programadores [18].

Como ponto primordial, antes de ser dado início ao desenvolvimento do código propriamente dito, é necessário a importação do SDK do facebook feito através de :

```
import com.facebook.FacebookSdk;  
import com.facebook.appevents.AppEventsLogger;
```

Bem como possibilitar o acesso à Internet. Este ponto é crucial, tanto para a situação atual que está a ser desenvolvida, como para o funcionamento das restantes funcionalidades da aplicação, que consistirão na reprodução de ficheiros que serão descarregados através da Internet.

Para tal é necessário adicionar o seguinte *uses-permission* ao manifesto da aplicação.

```
<uses-permission android:name="android.permission.INTERNET"/>
```

Dadas estas pequenas configurações essenciais para o funcionamento sem problemas da aplicação, foi efetuado o restante processo para concluir esta etapa da implementação prática.

Assim, é necessário configurar toda a parte visível dos componentes necessários e posteriormente configurar quais as suas ações. Começando por configurar o botão pelo qual será efetuado o *login* através do facebook, é necessário, no caso deste trabalho introduzir o código para tal no *layout* respetivo à classe MainActivity.class, denominado activity_main.xml, exemplificando seguidamente o código referido anteriormente :

```
<com.facebook.login.widget.LoginButton
    android:id="@+id/fb_login"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="95dp">
</com.facebook.login.widget.LoginButton>
```

Após a criação do botão no *layout* é necessário tornar o mesmo funcional. Foram definidas três ações, para três casos possíveis na altura do *login*:

- Quando há sucesso ao efetuar o *login*;
- Quando são introduzidos dados de início de sessão incorretos;
- Quando por ordem do utilizador, o processo de *login* é cancelado.

Antes da implementação destas possibilidades, surge a necessidade de permitir retornar *feedback* ao utilizador, para que em qualquer uma destas situações seja possível ao mesmo ter conhecimento do que acontece na altura do *login*. Para tal e retomando novamente ao activity_main.xml, foi definido uma visualização de texto, que é utilizada especificamente para o *feedback* referido anteriormente. Com este objetivo foi implementado o seguinte código:

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="50dp"
    android:text="Login Status"
    android:id="@+id/textView"
    android:gravity="center"
    android:textStyle="bold"/>
```

Continuando a implementação, por forma a chegar ao ponto dos casos possíveis do *login*, é necessário a declaração das variáveis que são usadas neste processos, como também indicá-las na função `onCreate()`, de forma a que esta possa carregar corretamente os respetivos *layouts*.

Para tal foi implementado o seguinte troço de código na classe `MainActivity.class`:

```
public class MainActivity extends AppCompatActivity {

    LoginButton loginButton;
    CallbackManager callbackManager;
    TextView textView;
    Intent i;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        FacebookSdk.sdkInitialize(getApplicationContext());
        setContentView(R.layout.activity_main);
        loginButton = (LoginButton) findViewById(R.id.fb_login);
        textView = (TextView) findViewById(R.id.textview);
        callbackManager = CallbackManager.Factory.create();
    }
}
```

Após este procedimento, foram implementadas as situações resultantes das diferentes ações por parte do utilizador na altura do *login*, assim para os casos que foram anteriormente mencionados temos respetivamente :

- Sucesso no *login*: O parâmetro `LoginResult` conterá o novo `AcessToken` e as respetivas permissões concedidas, sendo também mostrada uma visualização de texto com a informação de que o *login* foi executado com sucesso. Esta situação é abordada neste trabalho na secção seguinte, quando for abordada a parte da implementação do reprodutor, dada a necessidade de que após o *login*, a aplicação tem de "saltar" para a tela do reprodutor.
- Dados de início de sessão incorretos: será mostrado apenas a informação de que os dados para realizar o *login* não estão corretos.
- Cancelamento do processo de *login* por parte do utilizador: É simplesmente mostrada a informação de que o *login* foi cancelado.

Interpretando estas palavras em linguagem de programação temos a seguinte estrutura de código:

```
loginButton.registerCallback(callbackManager,
    new FacebookCallback<LoginResult>() {

        @Override
        public void onSuccess(LoginResult loginResult) {
            textView.setText("Login Executado \n" +
                loginResult.getAccessToken().getUserId()
                + "\n" + loginResult.getAccessToken().getToken());
        }

        @Override
        public void onCancel() {

            textView.setText("Login Cancelado");
        }

        @Override
        public void onError(FacebookException error) {
            textView.setText("Dados Errados");
        }
    });
```

Finalmente de forma a encaminhar os resultados do *login* ao callbackManager, que foi criado na função onCreate(), é apresentada a função onActivityResult() definida da seguinte forma:

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data){

    callbackManager.onActivityResult(requestCode, resultCode, data);
}
```


Sendo possível observar, o resultado atual da implementação prática descrita até este ponto do trabalho nas figuras seguintes.

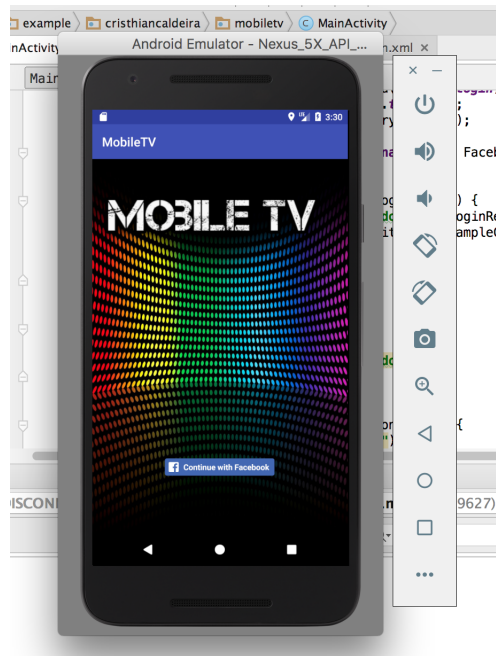


Figura 4.1: Página inicial da aplicação com o botão de *login* através do facebook.

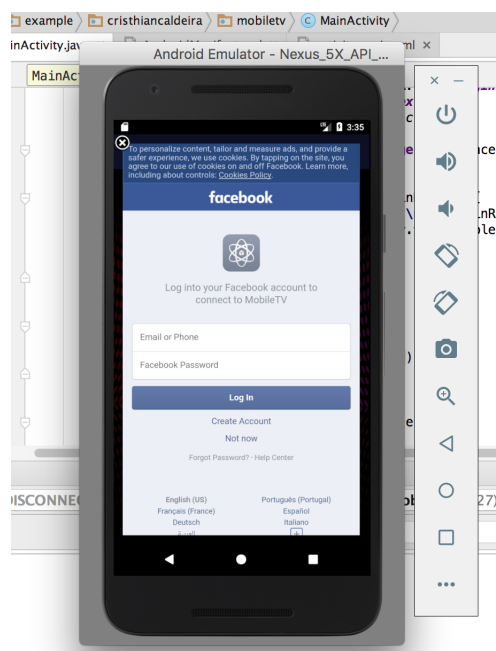


Figura 4.2: Pedido de dados para início de sessão através do facebook.

4.2 Reprodução de ficheiros DASH

Com o objetivo de proporcionar um reprodutor multimédia, capaz de reproduzir ficheiros MPEG-DASH, várias possibilidades surgiram, dado que o reprodutor nativo do Android não é compatível com os ficheiros DASH.

Assim, e após algum trabalho de pesquisa sobre o que já existe nesta campo e como criar algo novo para o mesmo efeito, foi decidido que a solução mais prática, eficiente e rentável, seria a implementação de um projeto já existente e disponibilizado pela Google sobre a forma de projeto *open source*, o ExoPlayer V2.

Dada esta possibilidade, foi necessário descarregar o projeto já existente e incorporá-lo no projeto da aplicação desenvolvido até então neste trabalho. É importante mencionar que todo o código referente ao reprodutor multimédia não foi elaborado neste trabalho nem é da autoria do autor do mesmo.

Desta forma, pequenas alterações foram executadas na implementação prática deste trabalho e como tal é apenas sobre elas que é falado nesta secção.

Feito o *download* do projeto através do próprio site destinado a programadores [19], deu-se a incorporação de todas as classes e respetivos *layouts* para o projeto desenvolvido até então para a concessão da aplicação final, sendo para tal incorporadas as seguintes classes:

- DemoApplication.class
- DemoUtil.class
- EventLogger.class
- PlayerActivity.class
- SampleChooserActivity.class
- TrackSelectionHelper.class

e os respetivos *layouts* existentes, como são o caso:

- list_divide.xml
- player_activity.xml
- sample_chooser_activity.xml
- track_selection_dialog.xml

Como é possível observar na imagem que se segue.

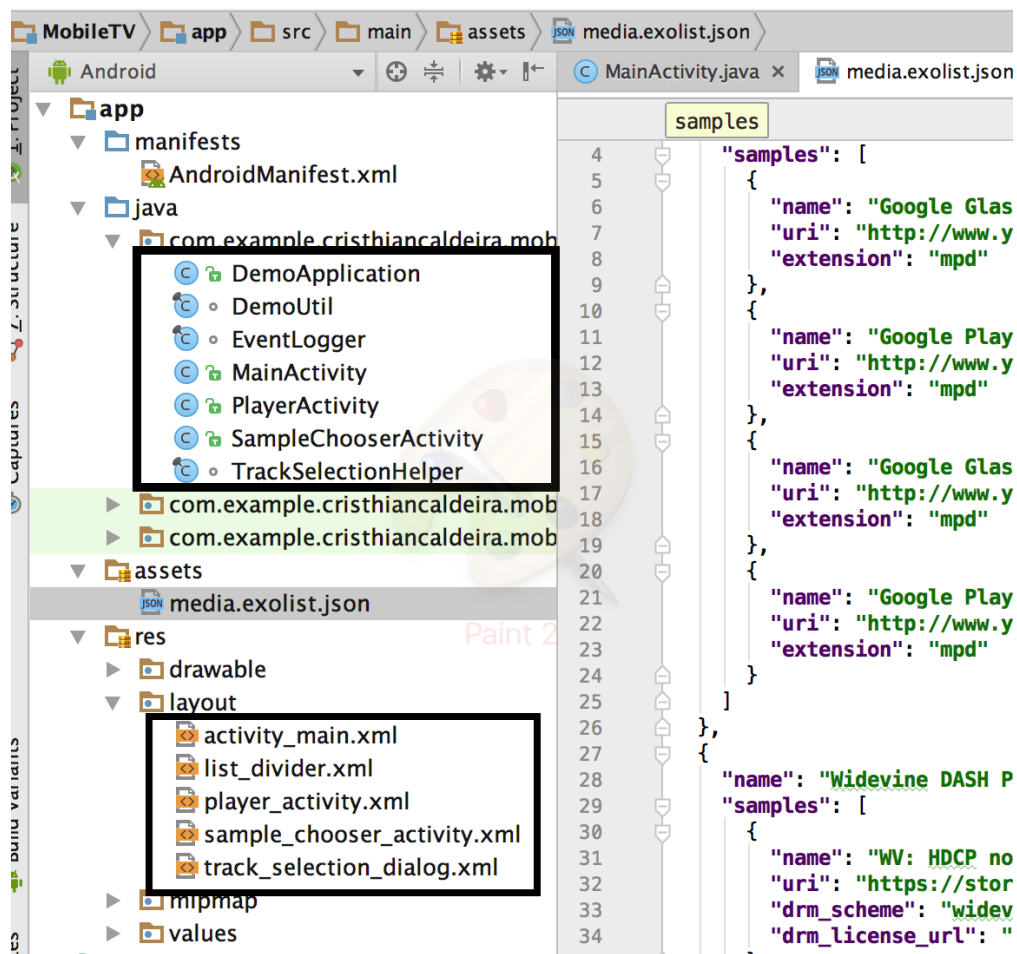


Figura 4.3: Classes e *layouts* incorporados do Projeto ExoPlayer.

Além da incorporação das classes e dos *layouts*, foi também necessário a incorporação do ficheiro *media.exolist.json*, que posteriormente é mencionado e explicada a sua relevância nesta aplicação. Mas antes disso, é necessário garantir dois aspetos, nomeadamente saber qual destas classes é a classe principal do reprodutor e após isso, garantir que o projeto desenvolvido até este ponto consiga fazer o salto para essa classe, integrando assim o ExoPlayer na aplicação que tem sido a ser desenvolvida.

Quanto ao primeiro ponto, através de análise do código e da documentação apresentada na pagina *developer guide* do ExoPlayer [20], foi possível ter conhecimento que a classe em questão é a *SampleChooserActivity.class*. Desta forma, fica apenas em falta um pequeno passo para incluir o reprodutor. Esta inclusão é garantida através da introdução do seguinte troço de código na *MainActivity.class*, na parte referente ao *login* com sucesso pelo facebook.

```
Intent i = new Intent(MainActivity.this, SampleChooserActivity.class);
startActivity(i);
```

Concretizadas estas implementações, é possível verificar que após efetuar o *login* com o facebook, a aplicação integra de forma correta o reprodutor necessário para a reprodução dos ficheiros DASH, como é visível na imagem que se segue.

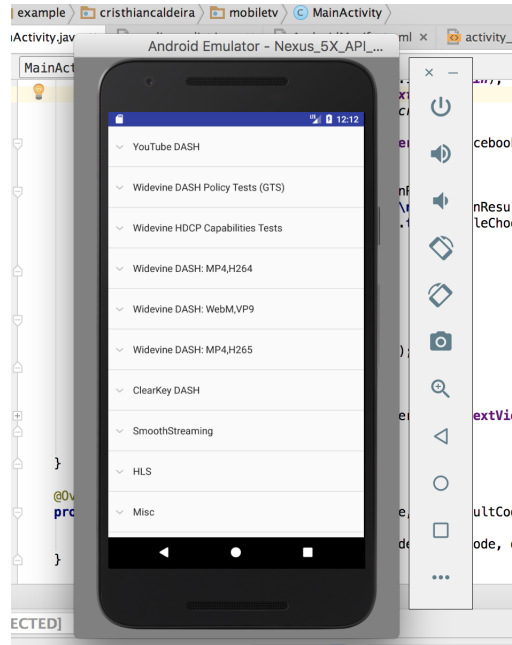


Figura 4.4: Tela de início do ExoPlayer após *login* com o facebook.

Como foi possível observar na imagem 4.4, a lista de conteúdos possível de reproduzir é a lista nativa do ExoPlayer. É neste ponto que é importante trabalhar o ficheiro `media.exolist.json`, dado que é através deste que é possível definir vários aspetos, nomeadamente:

- A quantidade de ligações disponíveis;
- O URL onde os conteúdos estão alocados;
- O título disponibilizado para a identificação dos mesmos;
- O tipo de extensão dos conteúdos, que neste caso são apenas ficheiros DASH.

4.3 Criação e Alocação dos ficheiros DASH

Por forma a codificar os ficheiros para o MPEG-DASH, é necessário certos passos de codificação e manipulação dos mesmo.

Inicialmente foi descarregado um ficheiro no formato mp4, através de uma página com exemplos de vídeo para *download* em diferentes formatos [21], que foi denominado `teste.mp4`. Com o objetivo de codificar este ficheiro dividindo-o em vários segmentos e gerando o MPD respetivo, foram seguidos um conjunto de etapas, descritas posteriormente nesta secção.

Em primeiro lugar, recorrendo à ferramenta MP4Box, foi efetuada uma codificação do ficheiro teste.mp4, criando vários segmentos de informação referente ao vídeo e áudio. Para este procedimento foi implementada a seguinte linha de código através da linha de comandos do computador.

```
MP4Box -dash 1000 -frag 1000 -rap -segment-name segment_ -out teste.mpd  
teste.mp4
```

Em que os parâmetros referidos representam:

- -dash 1000: Duração de cada segmento é de 1000 ms;
- -frag 1000: Criação de subsegmentos dentro de cada segmento, neste caso sendo o valor o mesmo, implica que cada segmento apenas contém um subsegmento;
- -rap: Força os segmentos a começar com um ponto de acesso aleatório;
- -segment-name segment_: Representa o nome atribuído aos segmentos, neste caso os segmentos serão denominados da seguinte forma: segment_1.m4s, segment_2.m4s, segment_3.m4s e assim sucessivamente;
- -out teste.mpd: Definir que o ficheiro MPD resultante, no final da codificação terá o nome teste.mpd, caso contrário assumiria por defeito o nome do ficheiro de entrada mais "_dash.mpd", neste caso seria teste_dash.mpd;
- teste.mp4: Ficheiro de entrada, a partir do qual serão efetuadas todas as manipulações necessárias para a codificação.

Dada a conclusão da codificação mencionada anteriormente, é necessário mais um passo por forma a ser possível reproduzir os ficheiros através de um reprodutor que reproduza ficheiros DASH. É necessário alocar os ficheiros resultantes da codificação num servidor HTTP.

Para tal e recorrendo aos serviços disponibilizados pela Faculdade de Engenharia da Universidade do Porto, esta alocação foi efetuada na pagina pessoal do autor deste documento:

<http://paginas.fe.up.pt/~ee09312> .

Desta forma e fazendo as seguintes alterações ao ficheiro media.exolist.json:

```
[
  {
    "name": "MobileTV",
    "samples": [
      {
        "name": "teste1",
        "uri": "https://paginas.fe.up.pt/~ee09312/teste1/teste.mpd",
        "extension": "mpd"
      }
    ]
  }
]
```

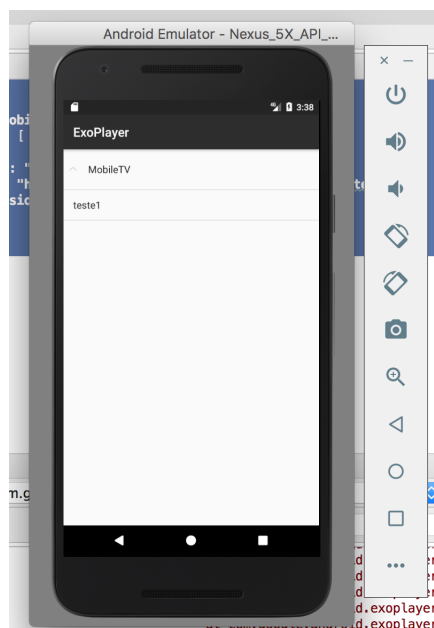


Figura 4.5: Tela da aplicação após as alterações referidas ao ficheiro media.exolist.json .

Estão reunias as condições para pôr em funcionamento a aplicação e conseguir visualizar o ficheiro codificado anteriormente.

No entanto, após ser testado o resultado obtido foi o seguinte:

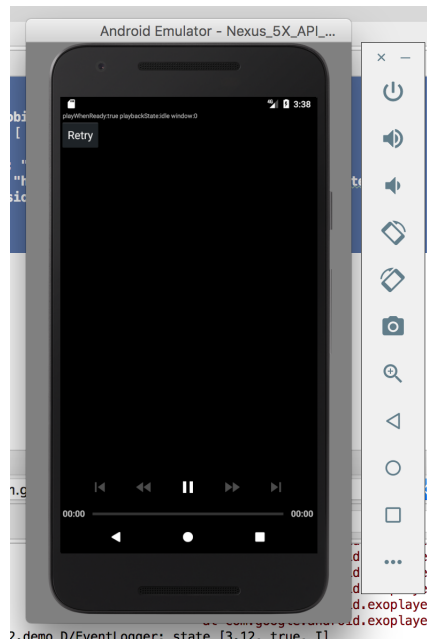


Figura 4.6: Tentativa falhada de visualizar o conteúdo codificado anteriormente .

Como é visível na figura 4.6, o reprodutor é iniciado com sucesso, mas a reprodução do ficheiro não.

Estando perante este problema, foi necessário voltar a pesquisar sobre a metodologia usada para a codificação e sobre o reprodutor utilizado para a visualização dos ficheiros. Assim, foi possível identificar que o problema ocorrente era fruto do modo de funcionamento do ExoPlayer, dado que as representações de áudio, vídeo e texto, para serem compatíveis com o reprodutor utilizado, precisam de estar codificadas separadamente.

Desta forma, uma nova metodologia de codificação foi usada, garantindo que as representações de áudio e vídeo estivessem separadas, pois neste caso não havia componente de texto no ficheiro a ser codificado. Esta metodologia foi implementada através da seguinte linha de código:

```
MP4Box -dash 1000 -frag 1000 -rap -segment-name segment_ -out teste.mpd  
teste.mpd#audio teste.mpd#video
```

Conseguindo desta forma obter um ficheiro MPD com representações de áudio e vídeo separadas.

Denotando-se que quando o ficheiro MPD possui as representações de áudio e vídeo juntas, estas podem ser traduzidas por:

```

...
<Representation id="1" mimeType="video/mp4" codecs="avc3.4D401F,mp4a.40.2"
width="1280" height="720" frameRate="25" sar="1:1" audioSamplingRate="48000"
startWithSAP="1" bandwidth="1421427">
  <AudioChannelConfiguration
    schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011"
    value="2"/>
...

```

Por outro lado, quando as representações contidas no ficheiro MPD estão separada, a estrutura escrita do mesmo já se traduz por:

```

...
<Representation id="1" mimeType="audio/mp4" codecs="mp4a.40.2"
audioSamplingRate="48000" startWithSAP="1" bandwidth="387029">
  <AudioChannelConfiguration
    schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011"
    value="2"/>
...
<Representation id="2" mimeType="video/mp4" codecs="avc3.4D401F"
width="1280" height="720" frameRate="25" sar="1:1" startWithSAP="1"
bandwidth="1035561">

```

Continuando a implementação e mudando o ficheiro media.exolist.json da forma descrita a seguir:

```

[
  {
    "name": "MobileTV",
    "samples": [
      {
        "name": "teste1",
        "uri": "https://paginas.fe.up.pt/~ee09312/teste1/teste.mpd",
        "extension": "mpd"
      },
      {
        "name": "teste2",
        "uri": "https://paginas.fe.up.pt/~ee09312/teste2/teste.mpd",
        "extension": "mpd"
      }
    ]
  }
]
}]

```


É possível constatar as seguintes alterações na pagina inicial do reprodutor da aplicação.

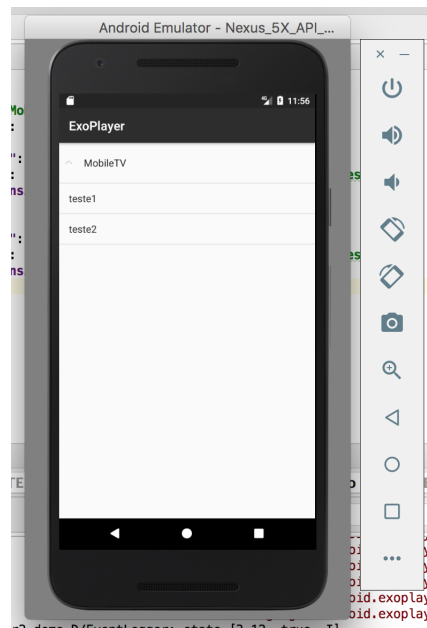


Figura 4.7: Nova tela da aplicação após a segunda alteração do ficheiro media.exolist.json.

Apesar de ser possível visualizar o conteúdo vídeo como, será possível constatar na imagem 4.8, surge um novo problema, no que diz respeito ao áudio, dado que apenas o vídeo é mostrado no reprodutor da aplicação, ou seja é possível visualizar um vídeo sem som.

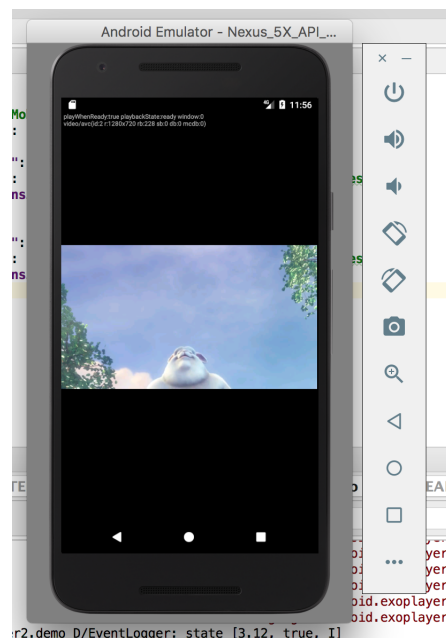


Figura 4.8: Tela da aplicação, sendo possível visualizar o vídeo mas não estando presente o áudio.

Foi feita uma análise aos ficheiros gerados durante a codificação através de :

```
MP4Box -dash 1000 -frag 1000 -rap -segment-name segment_ -out teste.mpd
teste.mpd#audio teste.mpd#video
```

Sendo possível observar que inicialmente são criados os segmentos de áudio e aquando da criação dos segmentos de vídeo, estamos perante uma situação de *overwrite*, dado que a instrução dada pela linha de código referida anteriormente, define que ambos terão o mesmo nome.

Caso o comando tivesse sido introduzido com uma ordem contrária, ou seja:

```
MP4Box -dash 1000 -frag 1000 -rap -segment-name segment_ -out teste.mpd
teste.mpd#video teste.mpd#audio
```

Primeiramente seriam criados os segmentos de vídeo e depois os segmentos de áudio, ocorrendo na mesma a situação de *overwrite*, sendo apenas possível ouvir e não visualizar o ficheiro resultante da codificação.

Uma possível resolução para este problema, é identificar cada representação (áudio e vídeo) e cada segmento, através de duas variáveis.

Assim, para uma correta codificação, foi alterada a linha de código adotada anteriormente, para a linha seguinte, incluindo as duas variáveis mencionadas anteriormente.

```
MP4Box -dash 1000 -frag 1000 -rap -segment-name
segment$RepresentationID$_$Number$ -out teste.mpd teste.mpd#audio
teste.mpd#video
```

Sendo o RepresentationID a variável que indica o numero da representação, neste caso é pertencente a [1;2], dado que apenas existem 2 representações (áudio e vídeo) e dada a ordem de execução, o áudio será representado pelo número 1 e o vídeo pelo número 2. Por sua vez a variável Number, representa o número do segmento.

Sendo um exemplo de um ficheiro gerado nesta codificação, o ficheiro segment_2_3.m4s, que é o segmento 3 da representação 2, neste caso o segmento 3 da representação de vídeo.

Desta forma, e alterando agora o ficheiro media.exolist.json da seguinte forma:

```
[
{
  "name": "MobileTV",
  "samples": [
    {
      "name": "teste1",
      "uri": "https://paginas.fe.up.pt/~ee09312/teste1/teste.mpd",
      "extension": "mpd"
    },
  ],
}
```

```

{
  "name": "teste2",
  "uri": "https://paginas.fe.up.pt/~ee09312/teste2/teste.mpd",
  "extension": "mpd"
},
{
  "name": "teste3",
  "uri": "https://paginas.fe.up.pt/~ee09312/teste3/teste.mpd",
  "extension": "mpd"
}
]
}
]

```

É perceptível que a tela inicial referente ao reproduzidor, voltou novamente a alterar, apresentando neste ponto a seguinte forma:

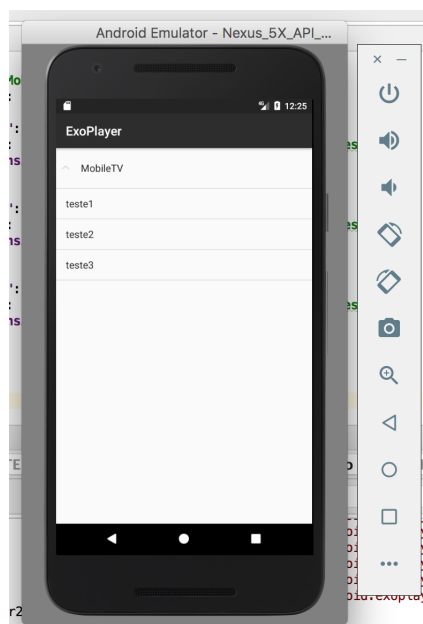


Figura 4.9: Nova tela da aplicação após a terceira alteração do ficheiro media.exolist.json.

Neste ponto de desenvolvimento do trabalho prático é já possível visualizar os conteúdos codificados segundo o DASH, através do reproduzidor da aplicação, como é possível observar na figura 4.10, estando neste caso presente a componente de áudio que estava em falta na codificação anterior, como é também possível de observar na figura 4.11.

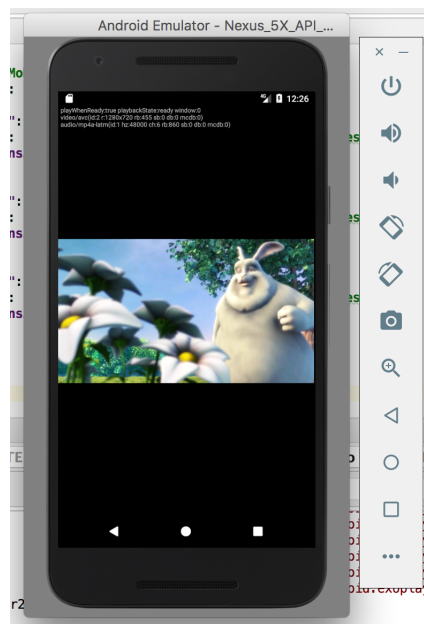


Figura 4.10: Tela da aplicação, sendo possível visualizar o vídeo.

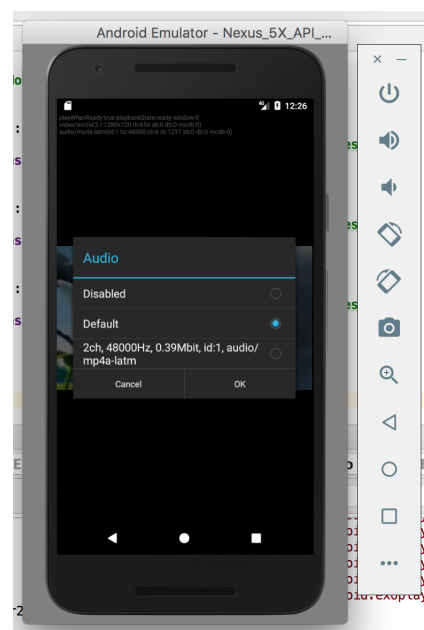


Figura 4.11: Tela da aplicação, sendo possível observar que a representação da componente de áudio esta presente.

Apesar de a codificação ter sido implementada e estar funcional, é necessário possibilitar diversas resoluções do mesmo ficheiro de vídeo, para tal e usando a ferramenta HamdBroke, foram gerados três ficheiros com o conteúdo do ficheiro teste.mp4, mas com diferentes resoluções, denominados de teste1028x720.mp4, teste 720x576.mp4 e teste70x30.mp4 sendo as suas resoluções respetivamente 1027x720, 720x576 e 70x30.

Dadas as pequenas dimensões dos dispositivos móveis e do emulador do Android Studio, usado neste trabalho, foi criado o ficheiro com resolução 70x30 por forma a ser perceptível a mudança de resolução entre as versões do vídeo, durante a sua reprodução.

Desta forma e pretendendo codificar os ficheiros anteriormente mencionados, foi usada a seguinte linha de código na linha de comando:

```
MP4Box -dash 1000 -frag 1000 -rap -bs-switching no -segment-mame
segment$RepresentationID$_$Number$ -out teste.mpd
teste1028x720.mp4#audio teste1028x720.mp4#video
teste720x576.mp4#audio teste720x576.mp4#video
teste70x30.mp4#audio teste70x30.mp4#video
```

Nesta codificação é necessário usar o parâmetro `-bs-switching` por forma a garantir que o ficheiro de Init é criado para as diferentes versões, quer para o áudio quer para o vídeo.

Dado isto o ficheiro `media.exolist.json` foi alterado para a seguinte forma:

```
[
  {
    "name": "MobileTV",
    "samples": [
      {
        "name": "teste1",
        "uri": "https://paginas.fe.up.pt/~ee09312/teste1/teste.mpd",
        "extension": "mpd"
      },
      {
        "name": "teste2",
        "uri": "https://paginas.fe.up.pt/~ee09312/teste2/teste.mpd",
        "extension": "mpd"
      },
      {
        "name": "teste3",
        "uri": "https://paginas.fe.up.pt/~ee09312/teste3/teste.mpd",
        "extension": "mpd"
      },
      {
        "name": "teste4",
        "uri": "https://paginas.fe.up.pt/~ee09312/teste4/teste.mpd",
        "extension": "mpd"
      }
    ]
  }
]
```

Sendo o resultado desta alteração visível na pagina inicial do reprodutor da aplicação, como se traduz pela seguinte imagem.

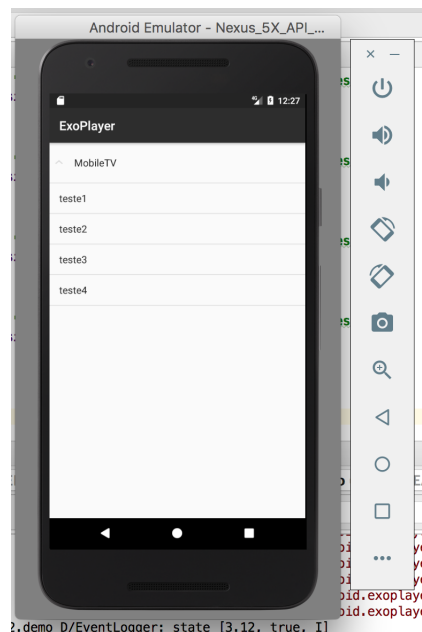


Figura 4.12: Nova tela da aplicação após a última alteração ao ficheiro media.exolist.json.

Finalmente é possível observar o conteúdo dos ficheiros DASH, através da aplicação, estando disponível vídeo em diversas resoluções e áudio.

A imagem 4.13 traduz uma situação onde inicialmente acontece uma reprodução do conteúdo com uma resolução de 1028x720, ou seja com uma boa qualidade, posteriormente acontece a mudança para uma resolução 70x30 e no final é possível observar uma visualização do conteúdo com a versão 70x30 carregada, que corresponde a uma visualização com má qualidade.



Figura 4.13: Exemplo de uma reprodução com boa qualidade, da mudança entre resoluções e de uma reprodução com má qualidade.

Capítulo 5

Conclusões e Trabalho Futuro

5.1 Conclusão

Após toda a pesquisa e trabalho efetuado para este projeto, foi possível retirar algumas ilações e conclusões, nomeadamente do tipo de tecnologia e metodologia mais eficaz, bem como ilações sobre o sucesso do serviço.

Dada a era moderna e tecnológica atual, foi possível concluir, que a melhor forma de implementar um serviço, como o abordado nesta dissertação é através da Internet e de todas as ferramentas existentes nela e para ela, não se justificando alguns dos custos necessários para a transmissão de sinal de televisão, para dispositivos móveis, associados às metodologias que exigem grandes infraestruturas, como as tecnologias abordadas no capítulo de Revisão Bibliográfica deste documento.

Desta forma surge como solução para este trabalho, a transmissão através do protocolo já existente e amplamente implementado na Internet, o HTTP.

Tendo consciência que diversos fatores influenciam o sucesso do serviço de *MobileTV*, e que através de HTTP *Streaming*, são várias as alternativas existentes, é crucial ponderar e optar pela que melhor se adapta a tais fatores.

Sabendo todos os fatores mencionados anteriormente, foi possível concluir, que a melhor adaptação é a transmissão através do MPEG-DASH, apesar de que em termos de características ser semelhante a outras formas de transmissão por HTTP, o facto de esta solução ser de acesso livre (não exigir equipamento ou sistema operativo específico), torna-a amplamente acessível, quer a nível computacional, quer a nível económico. Sendo também uma excelente solução para os produtores de conteúdos televisivos, pois este permite a difusão para diversas plataformas, possibilitando assim uma maior adaptabilidade aos dispositivos de receção, o que se pode tornar bastante vantajoso para quem já produz conteúdos e pretende transmiti-los a uma larga escala.

Numa reflexão socio-geográfica e económica acerca do serviço de *MobileTV*, é possível concluir que, em países com elevado poder de compra, o sucesso deste serviço é elevado, já o mesmo não acontece em países onde o poder económico é considerado médio/baixo.

Em suma, é possível afirmar que não está em questão o interesse pelo serviço, mas sim a possibilidade de aceder ao mesmo. Indo de acordo com o que anteriormente foi mencionado constata-se que, a melhor opção para esta problemática e por consequência para este trabalho, é uma opção de acesso livre e gratuito.

5.2 Trabalho Futuro

Sendo um dos focos desta problemática a interatividade com o utilizador, é inevitável após o desenvolvimento deste trabalho apontar como trabalho futuro um maior cuidado e investimento nesta área, dado que um pequeno passo possa ter sido dado neste trabalho, através da implementação do *login* com o facebook, várias outras formas de interação podem ainda ser introduzidas, enriquecendo toda a forma de interagir com o utilizador, sendo apenas um pequeno exemplo disso, implementar a possibilidade de partilhar em tempo real, através das redes sociais, a informação sobre o que o utilizador esta a assistir.

Em segundo lugar, mas talvez com um peso mais significativo para um trabalho futuro deste projeto, surge a possibilidade de tentar implementar a visualização de conteúdos que estejam a acontecer em tempo real, bem como tornar a seleção da resolução do vídeo automático, dependendo da largura de banda disponível. Sendo necessário para tal um estudo mais aprofundado dos diversos perfis do DASH, colmatando assim as limitações atualmente presentes na aplicação, que funciona para vídeos *On demand* e onde a seleção da resolução tem de ser feita manualmente pelo utilizador.

Referências

- [1] Televisão de alta definição resolução de ecrã. http://upload.wikimedia.org/wikipedia/commons/thumb/1/11/Standard_video_res.svg/500px-Standard_video_res.svg.png.
- [2] TDI operacoes stb. <https://image.slidesharecdn.com/tvdsserpro-120119160448-phpapp02/95/palestra-tv-digital-interatividade-mobil.jpg?cb=1326989700>. Accessed: 2017-03-01.
- [3] Padroes no Mundo televisao digital. <http://www.vcolor.com.br/nova/hdtv.bmp>. Accessed: 2016-05-27.
- [4] Thomas Stockhammer. Dynamic adaptive streaming over http-: standards and design principles. Em *Proceedings of the second annual ACM conference on Multimedia systems*, páginas 133–144. ACM, 2011.
- [5] Andrew Fechey-Lippens. A review of http live streaming. *Internet Citation*, páginas 1–37, 2010.
- [6] GENI adaptive video streaming. <https://witestlab.poly.edu/respond/sites/genitutorial/module/video-dash>. Accessed: 2017-04-03.
- [7] Iraj Sodagar. The mpeg-dash standard for multimedia streaming over the internet. *IEEE MultiMedia*, 18(4):62–67, 2011.
- [8] Gpac What is GPAC? . <https://gpac.wp.imt.fr/home/>.
- [9] Michael Löwy. As aventuras de karl marx contra o barão de münchhausen. *São Paulo: Busca Vida*, 1987.
- [10] Jorge Fernandes, Guido Lemos, e Gledson Silveira. Introdução à televisão digital interativa: arquitetura, protocolos, padrões e práticas. Em *Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação*, páginas 1–56, 2004.
- [11] Ondas eletromagnéticas propriedades ondas eletromagneticas. <http://mundoeducacao.bol.uol.com.br/fisica/o-que-sao-ondas-eletromagneticas.htm>. Accessed: 2017-02-25.
- [12] Angelo Brayner, Dorotéa Karine D Pitombeira, e Ricardo Wagner C Brito. Uma arquitetura eficiente para armazenamento, compressão e acesso a dados em dispositivos móveis com recursos computacionais limitados. Em *SBB*, páginas 250–264, 2005.
- [13] Ana Vitória Joly e João Carlos Massarolo. A interatividade na televisão digital-um estudo preliminar. *Universidade Federal de São Carlos*, 2001.

- [14] Estratégias de produção de motion graphics para mobile TV o contexto português. <http://biblioteca.versila.com/2758753>. Accessed: 2016-06-04.
- [15] Shin'ichiro Azuma, Hiroshi Kawamura, Shuichi Kawama, Shinji Toyoyama, Takao Hasegawa, Kanetomo Kagoshima, Masato Koutani, Hiroshi Kijima, Keiichi Sakuno, e Kunihiko Iizuka. A digital terrestrial television (isdb-t) tuner for mobile applications. Em *Solid-State Circuits Conference, 2004. Digest of Technical Papers. ISSCC. 2004 IEEE International*, páginas 278–528. IEEE, 2004.
- [16] MFLO what is "mediaflo"? <http://www.mobileburn.com/definition.jsp?term=MediaFLO>. Accessed: 2017-03-05.
- [17] Letícia Passos Affini e Elica Ito. Televisão digital móvel e portátil. *Revista GEMInIS*, (1 Ano 2):106–115, 2011.
- [18] facebook for developers. <https://developers.facebook.com>. Accessed: 2017-05-20.
- [19] Exoplayer Home. <https://google.github.io/ExoPlayer/>. Accessed: 2017-04-08.
- [20] Exoplayer Developer guide. <https://google.github.io/ExoPlayer/guide.html>. Accessed: 2017-04-10.
- [21] Download Sample Videos. <http://www.sample-videos.com>. Accessed: 2017-06-2.